N

Up to _Cryptography_

Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms

By David Chaum

Author's present address: Computer Science Division, Electrical
Engineering and Computer Sciences Department, University of
California, Berkeley, California 94720. (415)42-1024.

David L. Chaum
University of California, Berkeley

Abstract

A technique based on public key cryptography is presented that allows
an electronic mail system to hide who a participant communicates with
as well as the content of the communication - in spite of an unsecured
underlying telecommunication system. The technique does not require a
universally trusted authority. One correspondent can remain anonymous
to a second, while allowing the second to respond via an untraceable
return address.

The technique can also be used to form rosters of untraceable digital
pseudonyms from selected applications. Applicants retain the
exclusive ability to form digital signatures corresponding to their

pseudonyms.   Elections in which any interested party can verify that
the ballots have been properly counted are possible if anonymously
mailed ballots are signed with pseudonyms from a roster of registered
voters.   Another use allows an individual to correspond witha
record-keeping organization under a unique pseudonym which appears in
a roster of acceptable clients.

Key Words and Phrases: electronic mail, public key cryptosystems,
digital signatures, traffic analysis, security, privacy

CR Categories: 2.12, 3.81

Introduction

Cryptology is the science of secret communication.   Cryptographic
techniques have been providing secrecy of message content for
thousands of years [3].   Recently some new solutions to the "key
distribution problem" (the problem of providing each communicant with
a secret key) have been suggested [2,4], under the name of public key
cryptography.   Another cryptographic problem, "the traffic analysis
problem" (the problem of keeping confidential who converses with whom,
and when they converse), will become increasingly important with the
growth of electronic mail.   This paper presents a solution to the
traffic analysis problem that is based on public key cryptography.
Baran has solved the traffic analysis problem for networks [1], but
requires each participant to trust a common authority.   In contrast,
systems based on the solution advanced here can be compromised only by
subversion or conspiracy of all of a set of authorities.   Ideally,
each participant is an authority.

The following two sections introduce the notation and assumptions.
Then the basic concepts are introduced for some special cases
involving a series of one or more authorities.   The final section
covers general purpose mail networks.

Notation

Someone becomes a user of a public key cryptosystem (like that of
Rivest, Shamir, and Adleman [5]) by creating a pair of keys K and
Inv(K) from a suitable randomly generated seed.   The public key
K is made known to the other users or anyone else who cares to know
it; the private key Inv(K) is never divulged.   The encryption of
X with key K will be denoted K(X), and is just the image of X under
the mapping implemented by the cryptographic algorithm using key K.
The increased utility of these algorithms over conventional algorithms
results because the two keys are inverses of each other, ·in the sense
that

 Inv(K)(K(X)) = K(Inv(K)(X)) = X.

A message X is "sealed" with a public key K so that only the
holder of the private key Inv(K) can discover its content.   If
X is simply encrypted with K, then anyone could verify a guess
that Y = X by checking whether K(Y) = K(X).   This threat can be
eliminated by attaching a large string of random bits R to X
before encrypting.   The sealing of X with K is then denoted
K(R, X).   A user "signs" some material X by prepending a large
constant C (all zeros, for example) and then encrypting with its

private key, denoted Inv(K)(C, X) = Y. Anyone can verify that
Y has been signed by the holder of Inv(K) and determine the
signed matter X, by forming K(Y) = C, X and checking for C.

Assumptions

The approach taken here is based on two important assumptions:

(1) No one can determine anything about the correspondences between a
set of sealed items and the corresponding set of unsealed items, or
create forgeries without the appropriate random string or private key.

(2) Anyone may learn the origin, destination(s), and representation of
all messages in the underlying telecommunication system and anyone may
inject, remove, or modify messages.

Mail System

The users of the cryptosystem will include not only the correspondents
but a computer called a "mix" that will process each item of mail
before it is delivered. A participant prepares a message M for
delivery to a participant at address A by sealing it with the
addressee's public key Ka, appending the address A, and then
sealing the result with the mix's public key K1. The left-hand
side of the following expression denotes this item which is input to
the mix:

K1(R1, Ka(R0, M), A) --> Ka(R0, M), A.

The --> denotes the transformation of the input by the mix into the
output shown on the right-hand side. The mix decrypts its input with
its private key, throws away the random string R1, and outputs the
remainder. One might imagine a mechanism that forwards the sealed
messages Ka(R0, M) of the output to the addressees who then decrypt
them with their own private keys.

The purpose of a mix is to hide the correspondences between the items
in its input and those in its output. The order of arrival is hidden
by outputting the uniformly sized items in lexicographically ordered
batches. By assumption (1) above, there need be no concern about a
cryptanalytic attack yielding the correspondence between the sealed
items of a mix's input and its unsealed output - if items are not
repeated. However, if just one item is repeated in the input and
allowed to be repeated in the output, then the correspondence is
revealed for that item.

Thus, an important function of a mix is to ensure that no item is
processed more than once. This function can be readily achieved by a
mix for a particular batch by removing redundant copies before
outputting the batch. If a single mix is used for multiple batches,
then one way that repeats across batches can be detected is for the
mix to maintain a record of items used in previous batches. (Records
can be discarded once a mix changes its public key by, for example,
announcing the new key in a statement signed with its old private
key.) A mix need not retain previous batches if part of each random
string R1 constains something - such as a time-stamp - that is only
valid for a particular batch.

If a participant gets signed receipts for messages it submits to a
mix, then the participant can provide substantial evidence that the
mix failed to output an item properly. Only a wronged participant can

supply the receipt Y (= Inv(K1)(C, K1(R1, Ka(R0, M), A))), the
missing output X (= Ka(R0, M), A), and the retained string R1, such
that K1(Y) = C, K1(R1, X). Becasue a mix will sign each output
batch as a whole, the absence of an item X from a batch can be
substantiated by a copy of the signed batch.

The use of a "cascade", or series of mixes, offers the advantage that
any single constituent mix is able to provide the secrecy of the
correspondence between the inputs and the outputs of the entire
cascade. Incrimination of a particular mix of a cascade that failed
to properly process an item is accomplished as with a single mix, but
only requires a receipt from the first mix of the cascade, since a mix
can use the signed output of its predecessor to show the absence of an
item from its own input. An item is prepared for a cascade of n mixes
the same as for a single mix. It is then successively sealed for each
succeeding mix:

Kn(Rn, K<n-1>(R<n-1>, ... , K2(R2, K1(R1, Ka(R0, M), A))...)) -->.

The fist mix yields a lexicographically ordered batch of items, each
of the form

K<n-1>(R<n-1>, ... , K2(R2, K1(R1, Ka(R0, M), A))...) -->.

The items in the final output batch of a cascade are of the form
Ka(R0, M), A, the same as those of a single mix.

Return Addresses

The techniques just described allow participant x to send anonymous
messages to participant y. What is needed now is a way for y to
respond to x while still keeping the identity of x secret from y. A
solution is for x to form an untraceable return address
K1(R1, Ax), Kx, where Ax is its own real address, Kx is a public key
chosen for the occasion, and R1 is a key that will also act as a
random string for purposes of sealing. Then, x can send this return
address to y as part of a message sent by the techniques already
described. (In general, two participants can exchange return
addresses through a chain of other participants, where at least one
member of each adjacent pair knows the identity of the other member of
the pair.) The following indicates how y uses this untraceable return
address to form a response to x, via a new kind of mix:

K1(R1, Ax), Kx(R0, M) --> Ax, R1(Kx(R0, M)).

This mix uses the string of bits R1 that it finds after decrypting the
address part K1(R1, Ax) as a key to re-encrypt the message part
Kx(R0, M). Only the addressee x can decrypt the resulting output
because x created both R1 and Kx. The mix must not allow address
parts to be repeated - for the same reason that items of regular mail
must not be repeated. This means that x must supply y with a return
address for each item of mail x wishes to receive. Also notice that
conventional as opposed to public key cryptography could be used for
both encryptions of M.

With a cascade of mixes, the message part is prepared the same as for
a single mix, and the address part is as showin in the following
input:

K1(R1, K2(R2, ..., K<n-1>(R<n-1>, Kn(Rn, Ax))...)), Kx(R0, M) -->.

The result of the first mix is

K2(R2, ..., K<n-1>(R<n-1>, Kn(Rn, Ax))...), R1(Kx(R0, M)) -->,

and the final result of the remaining n-1 mixes is

Ax, Rn(R<n-1> ... R2(R1(Kx(R0, M)))...).

Untraceable return addresses allow the possiblity of "certified" mail:
They can provide the sender of an anonymous letter with a receipt
attesting to the fact that the letter appeared intact in the final
output batch. The address A that is incorporated into a certified
letter is expanded to include not only the usual address of the
recipient, but also an untraceable return address for the sender.
When this return address appears inthe output batch of the final mix,
it is used to mail the sender a signed receipt which inlucdes the
message as well as the address to which it was delivered. The receipt
might be signed by each mix.

Digital Pseudonyms

A digital "pseudonym" is a public key used to verify signatures made
by the anonymous holder of the corresponding private key. A "roster",
or list of pseudonyms, is created by an authority that decides which
applications for pseudonyms to accept, but is unable to trace the
pseudonyms in the completed roster. The applications may be sent to
the authority anonymously, by untraceable mail, for example, or they
may be provided in some other way.

Each application received by the authority contains all the
information required for the acceptance decision and a special
unaddressed digital letter (whose messages is the public key K, the
applicant's proposed pseudonym.) In the case of a single mix, these
letters are of the form K1(R1, K). For a cascade of n mixes, they
are of the form Kn(Rn, ..., K2(R2, K1(R1, K))...). The authority
will form an input batch containing only those unaddressed letters
from the applications it accepts. This input batch will be supplied
to a special cascade whose final output batch will be publically
available. Since each entry in the final output batch of the cascade
is a public key K from an accepted applicant, the signed output of the
final mix is a roster of digital pseudonyms.

Notification of applicants can be accomplished by also forming a
roster for unaccepted applications and then using the technique of
certified mail to return a single batch of receipts to both sets of
applicants. Of course, repeats must not be allowed within or across
batches.

If only registered voters are accepted for a particular roster, then
it can be used to carry out an election. For a single mix, each voter
submits a ballot of the form K1(R1, K, Inv(K)(C, V)), where K is
the voter's pseudonym and V is the actual vote. For a cascade of
mixes, ballots are of the form
Kn(Rn, ..., K2(R2, K1(R1, K, Inv(K)(C, V)))...). The ballots
must be processed as a single batch, as were the letters used to form
rosters. Items in the final lexicographically ordered output batch
are of the form K, Inv(K)(C, V). Since the roster of registered
voters is also ordered on K, it is easy for anyone to count the votes
by making a single pass through both batches at once. Each ballot is
counted only after checking that the pseudonym K which forms its
prefix, is also contained in the roster and that the pseudonym
properly decrypts the signed vote V.

An individual might be known to an organization only by a pseudonym
that appears in a roster of acceptable clients. Clients can
correspond with the organization via untraceable mail and the
organization can correspond with the clients using untraceable return
addresses. If applicants identify themselves in their applications,
or if they sign applications with pseudonyms that appear in a roster
issued by an authority that requires identification, then the
organization is assured that the same client cannot come to it under
different pseudonyms. Under special circumstances, such as default of
payment, a particular pseudonym could be shown to correspond to a
particular application (without revealing any other correspondences)
if each mix in turn supplied the appropriate Rn.

General Purpose Mail Systems

One way to construct a general purpose, untraceable mail system is to
require that every message pass through a cascade. Of course, mixes
can operate continuously or periodically, and long messages will be
encrypted first and then split into multiple items. In order to hide
the number of messages sent, each participant supplies the same number
of messages to each batch (some of which might be randomly addressed
dummies). In order to hide the number of messages received, each
participant privately searches the entire output for messages directed
to it.

Such a system may prove too costly for some participants. One way to
reduce the cost is to allow mail to be addressed to subsets of
participants, such as a local net. Participants that take advantage
of such arrangements need search only the mail addressed to a
particular subset. Another way to economize is for a participant to
send for each batch only the number of dummy messages suggested by a
random value (chosen from some suitable distribution), as opposed to
always sending the maximal number of messages. This can substantially
reduce message traffic and consequently, the size of output batches.
While these techniques may open the door to some kinds of statistical
attack, the system size that necessitated them may reduce the
effectiveness of such attacks.

In a large, general purpose mail system with many mixes, it may be
impractical for every message to pass through every mix. In such a
case, a sequence of mixes will be selected for each message, perhaps on
the basis of network topology or trust. Notice that if a participant
can choose mixes it trusts with its traffic volume data as early
members of its sequences, then these mixes can discard dummies they
receive from the participant and deliver small, fixed-sized batches
(padded with dummies) directly to the participant.

A new kind of mix will be presented here that allows a sequence of
mixes to be selected for each message. It also (a) hides the number
and identity of the mixes a message must pass through, (b) allows
incrimination of a mix that does not properly forward items, and (c)
makes no distinction between regular mail and mail sent by untraceable
return address. It is based on the idea that every item of mail is
composed of the same number of fixed-sized blocks.

The operations performed by this new kind of mix are always the same.
First it removes the first block and adds a random block J of junk to
the end, to maintain the item's length of l blocks. Then, using its
private key, the mix decrypts the block removed during the first step.
This yields a key R, which the mix uses to encrypt each of the l
blocks of the item (using either public-key or conventional
cryptography). It also yields the address A (either of a recipient or

of another mix) to which the item will be forwarded.

The left-hand side of the following shows how an item is prepared to
pass through a single mix:

```
K2( R2, ..., K<n-1>( R<n-1>, Kn( Rn, Ax ))...), R1( Kx( R0, M )) -->,

A1: [K<A1>( R<A1>, A )], [Inv(R<A1>)( M1 )], [Inv(R<A1>)( M2 )], ...
    [Inv(R<A1>)( M<l-1> )] --> A: [M1],...[M<l-1], [R<A1>( J<A1> )].
```

where square brackets show the extent of each block, and the sealed
message Ka(R0, M) is divided into pieces Mi, such that
Ka(R0, M) = M1, M2, ..., M<l-n>. The A1: indicates that the
left-hand side is delivered to mix A1, while the A: means that the
right-hand side is delivered to address A. Items with the same first
block should be regarded as repeats.

A message prepared to be passed through mixes A1 through An has the
form

```
A1: [K<A1>( R<A1>, A2 )], [Inv(R<A1>)( K<A2>( R<A2>, A3 ))], ...,
    [Inv(R<A1>)( Inv(R<A2>) ... Inv(R<A<n-1>>)( K<An>( R<An, A )) ... )],
    [Inv(R<A1>)( Inv(R<A2>) ... Inv(R<An>)( M1 )...)], ...,
    [Inv(R<A1>)( Inv(R<A2>) ... Inv(R<An>)( M<l-n> )...)]  -->.
```

The result leaving A1 is

```
A2: [K<A2>( R<A2>, A3 )], [Inv(R<A2>)( K<A3>( R<A3>, A4 ))], ...,
    [Inv(R<A2>)( Inv(R<A3>) ... Inv(R<A<n-1>>)( K<An>( R<An, A )) ... )],
    [Inv(R<A2>)( Inv(R<A3>) ... Inv(R<An>)( M1 )...)], ...,
    [Inv(R<A2>)( Inv(R<A3>) ... Inv(R<An>)( M<l-n> )...)],
    [R<A1>( J<A1> )] -->,
```

and the final result leaving An is

```
A: [M1], [M2], ..., [M<l-n>],
   [R<An>( R<A<n-1>> ... R<A1>( J<A1> )...)], ..., [R<An>( J<An> )].
```

An intermediate mix always knows which mix it received its input from -
by assumption (2) - but if a mix broadcasts copies of its fixed-size
output batches, then only individual recipient mixes need be able to
recognize their own input in a broadcast batch.

The untraceable return address x sends to y contains the key Kx that y
uses to encrypt the message part. It also includes, in the case of a
single mix, what y will use as the first block of the item it submits
to the mix:

```
A1: [K<A1>( R<A1>, Ax )], [M1], ..., [M<l-1>]  -->
Ax: [R<A1>( M1 )], ..., [R<A1>( M<l-1> )], [R<A1>( J<A1> )],
```

where Kx(R0, M) = M1, M2, ..., M<l-n>. Only x can decrypt the item
it receives since it created R<A1> and Kx. When a message is to pass
through n mixes, the untraceable return address contains the first n
blocks:

```
A1: [K<A1>( R<A1>, A2 )], [Inv(R<A1>)( K<A2>( R<A2>, A3 ))], ...,
    [Inv(R<A1>)( Inv(R<A2>) ... Inv(R<A<n-1>>)( K<An>( R<An>, Ax ))...)],
    [M1], [M2], ..., [M<l-n>]  -->.
```

After being operated on by mix A1 it will have the form

```
A2: [K<A2>( R<A2>, A3 )], ...,
```

```
        [Inv(R<A2>)( Inv(R<A3>) ... Inv(R<A<n-1>>)( K<An>( R<An>, Ax ))...)],
     .  [R<A1>( M1 )], [R<A1>( M2 )], ..., [R<A1>( M<1-n> )],
        [R<A1>( J<A1> )] -->,
```

and the final result leaving An is

```
Ax: [R<An>( R<A<n-1>> ... R<A1>( M1 )...)], ...,
    [R<An>( R<A<n-1>> ... R<A1>( M<1-n> )...)],
    [R<An>( R<A<n-1>> ... R<A1>( J<A1> )...)], ..., (R<An>( J<An> )].
```

Summary and Conclusion

A solution to the traffic analysis problem has been presented that allows any single intermediary to provide security for those messages passing through it. In addition, the solution allows messages to be sent or received pseudonymously. Through the notion of a roster of pseudonyms, it also provides some new and interesting kinds of limited anonymity.

References

1. Baran, P. On distributed communications: IX security secrecy and tamper-free considerations. Memo RM-3765-PR, Rand Corp., Santa Monica, CA, Aug. 1964.

2. Diffie, W. and Hellman, M.E. New directions in cryptography. IEEE Trans. Information Theory IT-22, 6 (Nov. 1976), 644-654.

3. Kahn, D. The Code Breakers, The Story of Secret Writing. Macmillan, New York, 1967.

4. Merkle, R.C. Secure communications over insecure channels. Comm. ACM 21, 4 (Apr. 1978), 294-299.

5. Rivest, R.L., Shamir, A., and Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. Comm. ACM 21, 2 (Feb. 1977), 120-126.